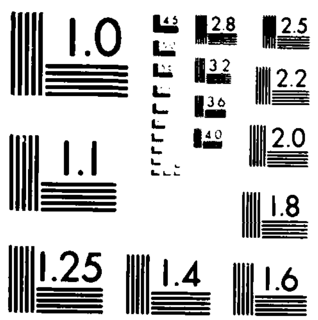
AD-A129 505 TECHNIQUE FOR CALCULATING POWER IN BROADBAND WAVEFORMS 1/1
USING SPECTRAL ANA..(U) ARMY MISSILE COMMAND REDSTONE
ARSENAL AL TEST AND EVALUATION D.. O T HOOKS ET AL.
UNCLASSIFIED OCT 82 DRSMI/RT-83-1-TR SBI-AD-E950 386 F/G 20/14 NL

END
DATE
FILMED
7 83
DTIC



MICROCOPY RESOLUTION TEST CHART
NATIONAL BUREAU OF STANDARDS 1963-A

ADA129505

TECHNICAL REPORT RT-83-1

TECHNIQUE FOR CALCULATING POWER IN BROADBAND WAVEFORMS USING SPECTRAL ANALYSIS

Orlando T. Hooks and Robert A Snead
Test and Evaluation Directorate
US Army Missile Laboratory

October 1982

U.S. ARMY MISSILE COMMAND
Redstone Arsenal, Alabama 35809

Approved for public release; distribution unlimited.

SMI FORM 1021, 1 JUL 79 PREVIOUS EDITION IS OBSOLETE

DISPOSITION INSTRUCTIONS

DISCLAIMER

TRADE NAMES

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

REPORT DOCUMENTATION PAGE

READ INSTRUCTIONS BEFORE COMPLETING FORM

1. REPORT NUMBER: RT-83-1	2. GOVT ACCESSION NO.: AD-A129 505
3. RECIPIENT'S CATALOG NUMBER	
4. TITLE (and Subtitle): Technique for Calculating Power in Broadband Waveforms using Spectral Analysis	5. TYPE OF REPORT & PERIOD COVERED: Technical Report
	6. PERFORMING ORG. REPORT NUMBER
7. AUTHOR(s): Orlando T. Hooks and Robert A. Snead	8. CONTRACT OR GRANT NUMBER(s)
9. PERFORMING ORGANIZATION NAME AND ADDRESS: Commander, US Army Missile Command, ATTN: DRSMI-RTS, Redstone Arsenal, AL 35898	10. PROGRAM ELEMENT, PROJECT, TASK AREA & WORK UNIT NUMBERS
11. CONTROLLING OFFICE NAME AND ADDRESS: Commander, US Army Missile Command, ATTN: DRSMI-RPT, Redstone Arsenal, AL 35898	12. REPORT DATE: 20 Sep 82
	13. NUMBER OF PAGES: 16
14. MONITORING AGENCY NAME & ADDRESS(if different from Controlling Office)	15. SECURITY CLASS. (of this report): UNCLASSIFIED
	15a. DECLASSIFICATION/DOWNGRADING SCHEDULE

16. DISTRIBUTION STATEMENT (of this Report)

Approved for public release; distribution unlimited.

17. DISTRIBUTION STATEMENT (of the abstract entered in Block 20, if different from Report)

18. SUPPLEMENTARY NOTES

19. KEY WORDS (Continue on reverse side if necessary and identify by block number)

Electro-explosive devices (EED)
Electromagnetic interference (EMI)
Spectral amplitude
Resolution bandwidth (RBW)

20. ABSTRACT (Continue on reverse side if necessary and identify by block number)

This report presents a worst case technique for measuring the power contained in a broad band waveform. (author

DD FORM 1 JAN 73 1473 EDITION OF 1 NOV 65 IS OBSOLETE

UNCLASSIFIED

SECURITY CLASSIFICATION OF THIS PAGE (When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

SECURITY CLASSIFICATION OF THIS PAGE(When Data Entered)

CONTENTS

I. INTRODUCTION

One of the dangers of using electro-explosive devices (EED), whether for military, commercial, or aerospace use, is unintentional detonation of the device due to electromagnetic energy. In order to evaluate this hazard, several techniques and algorithms have been developed, mostly for single frequency electromagnetic sources.

The development of the microcomputer and the drastic increase in the incidence of digital implementation which followed created additional problems in the area of electromagnetic interference (EMI). The interference emitted from a computing system is not limited to a single frequency, but instead falls into a broad spectrum, often covering several hundred megahertz (MHz). The designer using EED's must evaluate the power contained in the broadband emissions in order to insure that EED's used in close proximity to computing equipment do not receive enough energy to accidentally detonate. This report presents a "quick and dirty" approach to evaluate the worst case energy coupled into an EED. The actual energy received will always be less, due to assumption made in the algorithm.

The equipment required to use this method consists of a Hewlett-Packard 8568, a spectrum analyzer, and a computing controller equipped with an IEEE 488 bus interface.

Communications engineers like to express the frequency content of a time domain waveform v(t) in terms of a frequency domain spectral density g(f). This spectral density is effectively a measure of the power or voltage in each small frequency component of the signal. The total power contained in the waveform may be found by integrating the spectral power density of the waveform over the frequency range of interest. Mathematically;

$$P = \int_{f_1}^{f_2} G(f)df \tag{1}$$

If the waveform frequency domain data contains a series of discrete frequency components of amplitude A, then integral 1 becomes a discrete series summation.

$$P = \sum_{f_1}^{f_2} A_f \Delta f \tag{2}$$

If the bandwidth of the receiver is chosen carefully so that the power contained in one small increment of the frequency spectrum does not "spill over" into other increments (Δfs), then equation 2 allows one to find the total power contained in each increment of frequency.

The Hewlett-Packard 8568A and 8566B Spectrum Analyzers are scanning filter devices which presents the frequency domain data as a 1000 point array of vertical amplitudes over a linear range of frequency steps, so that Δf, the analyzer's frequency step satisfies equation 3.

$$\Delta f = \frac{f\ stop - f\ start}{1000} \tag{3}$$

If the analyzer's resolution bandwidth (RBW) is chosen less than Δf ($0.1\Delta f$ or less is ideal), each spectral amplitude, A will be influenced very little by the signal amplitudes on either side and the requirements of equation 2 are met $\{f_o + (M-1)\Delta f, A_{n-1}$ and $f_o + (M+1)\Delta f, A_{n+1}\}$. Since the HP8568A and 8566B allow the user to preselect the RBW, the power contained in a broadband waveform may be calculated easily by summing the amplitudes of each spectral component, assuming the user selected the RBW properly. This may also be accomplished under software control of an IEEE 488 bus compatible calculator.

The system this lab uses is composed of an HP8528A Spectrum Analyzer connected to an HP9825 Computing Controller using the HP Interface Bus (HPIB). A flowchart of the program used is shown in Figure 1.

The basic algorithm used to calculate the broadband power is very straightforward.

1. Set up the spectrum analyzer to take the emissions data recorded as dBµV.

2. Apply a frequency dependent correction factor for the antenna effects to convert each data point to dBµV/m

3. Convert E from its logrithmic form to its linear form

$$E(v/m) = 10^{\frac{E(dB\mu v/m)-6}{20}}$$

4. Convert E (v/m) to power density $P_d(W/m^2)$

$$P_d = E^2/377\,(W/m^2)$$

5. Calculate the worst case power coupled into an EED (due to each frequency component) assuming no losses due to matching, and reception of signal due to a resonant dipole.

$$p = \frac{Pd\ (1.65)\ \lambda^2}{4\pi}$$ where λ is the wavelength.

6. Sum each of the powers due to the individual components to find the total power.

II. PROGRAM DESCRIPTIONS

This program is divided into two sections. The first one (lines 0 - 21) is a loop which controls the program flow by calling appropriate subroutines. This section also initializes the computational and data acquisition equipment. The second section (lines 22-207) contains the subroutines called by the control section.

Each of the subroutines will be explained in detail in the following sections but some words of clarification are needed concerning the control section. Line 3 clears the HPIB interface and Line 19 stores the plotter pen.

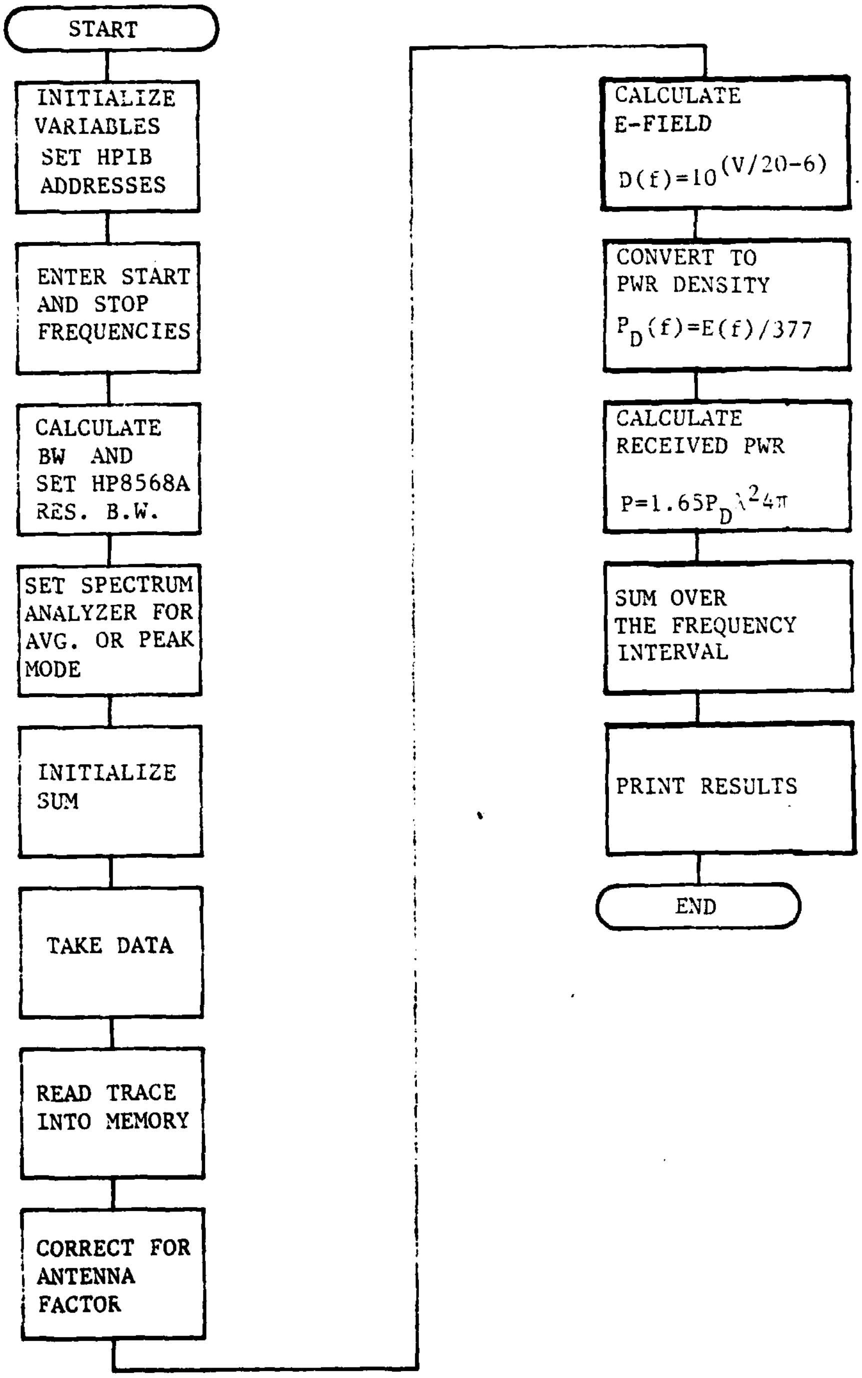


Figure 1. Flowchart

```
0: "  15 MAY 81  POWER SUMMATION  ":
1: "   POWSUM      ":
2: gsb "DEV"
3: cli 7
4: gsb "PLINIT"
5: gsb "ARRAY"
6: gsb "grat"
7: gsb "TITLE1"
8: gsb "title-out"
9: gsb "SETUP"
10: gsb "FREQ"
11: gsb "BWP"
12: gsb "SWEEP"
13: gsb "REDRL"
14: gsb "REDSA"
15: gsb "ANTCOR"
16: gsb "SUMMATION"
17: gsb "LEARN"
18: gsb "STATEOUT"
19: oen$
20: dsp "END OF TEST";beep;end
21:
22: "******************** SUBROUTINES ***************************":
23:
24: "DEV ASSIGNS THE HPIB ADDRESSES":
25: "DEV":dev "sa",718,"lp",6,"gr",705,"25",719,"ptr",714;ret
```

III. SUBROUTINE DESCRIPTIONS

A. DEV

Subroutine DEV uses the HPL device statement (DEV) to associate string variables with peripheral logical unit numbers (HPIB addresses). Line 25 makes the following assignments:

Device	Name	HPIB Address
HP8568 Spectrum Analyzer	sa	718
HP8966B Line Printer	lp	6
HP9872A Plotter	gr	705
HP9825 Controller	25	719
HP2631A Line Printer	ptr	714

The main reason for using the device statement is to facilitate program modification if the HPIB device address assignments must be changed. If peripheral devices were referenced directly by address, changing an HPIB address would mean modifying every statement which referenced that peripheral. By using the device statement this situation is averted, and only the device statement making the assignment must be changed if HPIB addresses are changed.

B. PLINIT

This routine initializes (clears) the 9872A Graphics Plotter and sets the plotting window scale points P1 and P2. It also sets the minimum and maximum scale values in user units.

```
95: "PLINIT":
96: "INITILIZES THE PLOTTER AND SETS THE VALUE OF P1 & P2":
97: wrt "gr","IN"
98: wrt "gr","IP 100,1000,13000,8000"
99: scl -100,1100,-20,120
100: ret
```

C. ARRAY

ARRAY dimensions the necessary storage arrays for the real and string variables. The HPL dim statement is equivalent to the FORTRAN DIMENSION statement. However, there is one difference. In HPL the string variables must be dimensioned. The statement dim A$(2), dimensions a string variable, 2 characters long to be named A$.

```
26: "ARRAY":
27: "ARRAY DIMENSIONS ALL ARRAYS":
28: dim X[100],Q[100],L[1001],A$[2],T$[64],D$[8,64]
29: ret
```

D. GRAT

The purpose of this subroutine is to draw one plot on which the final data will be plotted. Line 109 determines the size and tilt of the labeling. Lines 111 through 118 draw the horizontal lines representing the linear dBuV scale. The vertical grid lines represent the linear frequency scale.

```
107: "grat":
108: "draws the graph on the plotter":
109: csiz 1,2,1,0
110: pen# 1
111: plt 0,0,-2
112: for I=0 to 90 by 20
113: iplt 1000,0
114: iplt 0,10
115: iplt -1000,0
116: iplt 0,10
117: next I
118: iplt 1000,0
119: pen
120: plt 0,0,-2
121: for I=0 to 400 by 100
122: iplt 0,100
123: iplt 100,0
124: iplt 0,-100
125: iplt 100,0
126: next I
127: iplt 0,100
128: pen
129: ret
```

E. TITLE1 and TITLE-OUT

TITLE1 inputs an array of string variables which serves as a graph data identification block. The input format is eight string variables, each 64 characters long. TITLE-OUT simply prints the strings input in TITLE1 on the line printer.

```
84: "TITLE1":
85: "INPUT THE DOCUMENTATION INFORMATION TO BE PLACED ON THE PLOTS":
86: ent "SYSTEM NAME?",T$;"SYSTEM NAME: "&T$→D$[1];""→T$
87: ent "TEST DATE?",T$;"TEST DATE: "&T$→D$[2];""→T$
88: "MICOM/DRSMI-RTR"→D$[3]
89: ent "YOUR NAME?",T$;"NAME: "&T$→D$[4];""→T$
90: ent "TEST NUMBER?",T$;"TEST NUMBER: "&T$→D$[5];""→T$
91: ent "MODE?",T$;"MODE: "&T$→D$[6];""→T$
92: ent "POLARIZATION?",T$;"POLARIZATION: "&T$→D$[7];""→T$
93: ent "TEST CONFIGURATION?",T$;"TEST CONFIGURATION: "&T$→D$[8];""→T$
94: ret
```

```
130: "title-out":
131: "outputs the annotation strings entered by the operator":
132: wrt "ptr",char(10)&char(13)
133: wrt "ptr",char(27)&"n"
134: for I=1 to 8
135: wrt "ptr",D$[I]
136: next I
137: wrt "ptr",char(27)&"o"
138: ret
```

F. SETUP

This subroutine sets up the spectrum analyzer to collect the data using the following commands:

<u>IP</u> - Run the spectrum analyzer instument preset routine. Set all controls and functions to their default states.

<u>KSC</u> - Set the display units to dBμV.

<u>ATODB</u> - Set the spectrum analyzer input attenuation to 0 db.

<u>I1</u> - Select input 1. The HP8568A has two signal inputs. Input 1 is selected because it has a wider bandwidth (100 Hz - 1.5 GHz) than input 2 (100 KHz - 1.5 GHz).

```
30: "SETUP":
31: "SETS THE 8568A REFERENCE LEVEL":
32: wrt "sa","IP KSC"
33: wrt "sa","ATODB"
34: ret
```

G. FREQ

This subroutine passes the start and stop frequencies for the band to be displayed to the spectrum analyzer.

```
35: "FREQ":
36: "FREQ SELECTS THE FREQUENCY SPAN OF THE ANALYZER":
37: wrt "sa","FB200MZ,FA20MZ";ret
```

H. BWP

This subroutine passes the video and resolution bandwidth parameters to the spectrum analyzer. The resolution and video bandwidths in this program represent an optimum set of parameters arrived at by balancing the following requirements:

1. The noise floor of the instrumentation system is required to be at least 6 dB below the MIL-STD 461 limits.

2. The sweep time of the spectrum analyzer display must be appropriate. Under some combinations of parameters, the sweep times may be as long as 25 minutes; much too long to run repetitive tests.

```
38: "BWP":
39: "BWP PASSES THE BANDWIDTH PARAMETERS TO THE 8568A":
40: wrt "sa","RB10KZ,VB10MZ"
41: ret
```

I. SWEEP

This subroutine triggers the spectrum analyzer sweep, using the A trace memory (trace B is blanked). The trace is cleared and the max hold (peak) mode is set. The spectrum analyzer is allowed to sweep continuously for approximately 20 seconds. This helps to pick up transient noise with a low duty cycle.

```
42: "SWEEP":
43: "SWEEP CAUSES THE 8568A TO SWEEP THE BAND & DISPLAY IT":
44: wrt "sa","HD A1"
45: wrt "sa","A2"
46: wrt "sa","S2"
47: wrt "sa","TS"
48: wrt "sa","A3"
49: ret
```

J. REDRL

The sole purpose of this subroutine is to read the current reference level from the spectrum analyzer and store it in variable R for later use.

```
101: "REDRL":
102: "READ THE 8568A REFERENCE LEVELS":
103: wrt "sa","RL OA"
104: red "sa",R
105: wrt "sa","HD"
106: ret
```

K. REDSA

REDSA is the subroutine which reads the raw data stored in the spectrum analyzer trace memory into the 9825 controller for analysis.

First the start and stop frequencies are read using the OA (output active function) command. The start frequency is stored in r2 and the stop frequency is stored in r3; both are used in later subroutines.

The trace data is then read as a 1000 element array of vertical points, with the element number representing the X coordinate and the vertical magnitude representing the Y coordinate. The displayed magnitudes range from a minimum of 0 to a maximum of 1040 in screen coordinates. This creates a square field of points, each side of which is 1000 points in length, which leads to the Cartesian coordinate system explained above.

The FOR-NEXT loop contained in lines 61-64 is the portion of the program which transfers the raw data from the spectrum analyzer to the controller.

Line 63 is of particular interest. The first portion of the line divides each element of the raw data array (L[I]) by 1000 and multiplies by 10 to convert the screen magnitude fo dB referenced to the lowest line on the screen graticle. The next portion of line 63 (R-100) establishes the reference level for the top line of the graticle (using subroutine REDRL) and translates that reference level to the bottom line of the screen graticle. The two numbers are then added, producing an absolute reading in dBuV.

At this point, the data stored in L[I] should correlate with the raw data read on any EMI receiver, if the bandwidths are the same, the data-point frequencies are the same, and the instrument tolerances are accounted for.

```
50: "REDSA":
51: "REDSA READS THE TRACE DATA FROM THE 8568A":
52: "first read the start and stop frequencies":
53: fmt
54: wrt "sa","FA OA"
55: red "sa",T;T→r2
56: wrt "sa","FB OA"
57: red "sa",T;T→r3
```

```
58: wrt "sa","4D"
59: "now read the data in screen (X,Y) coordinates":
60: wrt "sa","O1 TA"
61: for I=1 to 1001
62: red "sa",L[I]
63: L[I]/1000*100+(R-100)→L[I]
64: next I
65: ret
```

L. ANTCOR AND ANTCORLIM

These subroutines are closely related and will be considered at the same time. ANTCOR takes the data array output from CHOP or BCOR and corrects for the effects of the antenna factor. The corrected data is plotted on the graph generated in GRAT. Due to the complexity of ANTCOR, it will be explained in detail.

Line 68 calculates the frequency difference between sampling points by subtracting the start frequency (r2) from the stop frequency (r3) and dividing by the number of points.

Line 72 calculates the frequency of each measurement point and converts to an integral number of MHz.

Line 73 calls ANTCORLIM, which supplies the slope and intercept to generate a log-linear function used to approximate the frequency currently contained in r1. (The approximation will be explained in detail later).

Line 74 initializes r6.

Line 76 calculates the antenna correction factor.

Line 77 adds the antenna factor to the data.

Lines 79 and 80 actually plot the corrected data [each datapoint of L (I)].

Line 79 is necessary to plot the first point in each range.

Line 80 plots all others.

ANTCORLIM contains a set of variables used to generate a linear approximation to the antenna factor. These variables were calculated using the following procedure:

```
66: "ANTCOR":
67: "THIS ROUTINE CALCULATES THE FREQUENCY DELTA":
68: r3-r2/1000→r4
69: pen# 1
70: "to plot data":
71: for I=1 to 1001
72: (r2+(I-1)*r4)/1e6→F
```

```
73: gsb "ANTCORLIM"
74: 0→r6
75: "now generate the polynomial approximation to the antenna factor":
76: S*log(F*1e6)+A→r6
77: r6+L[I]→L[I]
78: if I=1;plt I,L[I],-2;gto +2
79: if I=1;plt I,L[I],-2
80: plt I,L[I]
81: next I
82: pen
83: ret
```

1. The antenna correction factors given by the manufacturer were plotted on semi-log paper.

2. The curve on the log chart was then approximated by a series of straight lines over frequency intervals.

3. The starting and stopping frequencies for each line were noted, and the slope and intercept of each linear segment were calculated using the following relationships:

$$S = \frac{4 - r2}{\log(r3) - \log(r1)} \qquad I = r2 - S \log (r1)$$

r1 = interval starting frequency
r2 = starting amplitude
r3 = stop frequency
r4 = stop amplitude
S = slope
I = Intercept

These slopes and intercepts then enable the calculation of the antenna factor for any datapoint if the frequency is known. The frequencies were calculated in line 72 using the semilog relationship:

$$Y = m \log x + b$$

```
139: "ANTCORLIM":
140: if F>=2e1 and F<3e1;-1.135774717→S;1.929232528e1→A;ret
141: if F>=3e1 and F<4e1;2.5625529e1→S;-1.807081637e2→A;ret
142: if F>=4e1 and F<5e1;-2.063770232e1→S;1.708890511e2→A;ret
143: if F>=5e1 and F<6e1;-3.409898347e1→S;2.745270509e2→A;ret
144: if F>=6e1 and F<7e1;-1.941840687e1→S;1.603393056e2→A;ret
145: if F>=7e1 and F<8e1;1.034626412e1→S;-7.316745634e1→A;ret
146: if F>=8e1 and F<9e1;1.954937807e1→S;-1.459004941e2→A;ret
147: if F>=9e1 and F<1e2;1.180134647e2→S;-9.291077175e2→A;ret
148: if F>=1e2 and F<1.1e2;-2.657474373e1→S;2.275979498e2→A;ret
149: if F>=1.1e2 and F<1.2e2;2.381673128e1→S;-1.776196887e2→A;ret
```

```
150: if F)=1.2e2 and F(1.3e2;-2.876695653+S;3.304134557e1+A;ret
151: if F)=1.3e2 and F(1.4e2;-6.835549619e1+S;5.693326239e2+A;ret
152: if F)=1.4e2 and F(1.5e2;6.007364352e1+S;-4.768675917e2+A;ret
153: if F)=1.5e2 and F(1.6e2;4.638099184e1+S;-3.649152219e2+A;ret
154: if F)=1.6e2 and F(1.7e2;6.456773725e1+S;-5.141214634e2+A;ret
155: if F)=1.7e2 and F(1.8e2;2.819899046e1+S;-2.147903506e2+A;ret
156: if F)=1.8e2 and F(1.9e2;-5.110494019e1+S;4.398852076e2+A;ret
157: if F)=1.9e2 and F(2e2;9.875924846e1+S;-8.008034838e2+A;ret
158: ret
159: "SUMMATION":
160: 0+J
161: fmt 6,38x,e12.5
162: fmt 5,18x,"**********SUMMATION OF POWER(WATTS/SQUARE METER********"
163: for I=1 to 1001
164: L[I]+V
165: tnf(V/20-6)+P
166: P↑2/377+P
167: gto +2
168: 3e8/F+L
169: P+D
170: gsb "SUM"
171: next I
172: wrt 6.5
173: wrt 6.6,J
174: ret
175: "*********SUBROUTINES**********":
176: "SUM":
177: D+J+J
178: ret
179: "LEARN":
180: "THIS SUBROUTINE SAVES THE STATE OF THE INSTRUMENT":
181: dim C$[11,15]
182: wrt 718,"OT"
183: for N=1 to 11
184: red 718,C$[N]
185: next N
186: ret
187: "STATEOUT":
188: for I=1 to 11
189: prt C$[I]
190: next I
191: plt -100,-5,-2
192: lbl C$[10]
193: plt 0,-10,-2
194: lbl C$[3]
195: plt 450,-10,-2
196: lbl C$[4]
197: plt 725,-10,-2
198: lbl C$[5]
199: plt 995,-5,-2
200: lbl C$[11]
201: plt -100,95,-2
202: lbl C$[8]
```

```
203: plt -100,102,-2
204: lbl "HP"
205: plt -40,102,-2
206: lbl C$[7]
207: plt 240,102,-2
208: lbl C$[6]
209: ret
*6335
```

M. SUMMATION

Subroutine SUMMATION obtains the power contained in each incremental frequency division and sums them over the band of interest. This is done in the following manner.

Line 160 sets the sum counter J = 0.

Line 161 is a format statement that causes the line printer to skip 38 spaces and print out a 12 digit number with 5 digits to the right of the decimal point.

Line 162 sets up the format for the label "*** SUMMATION OF POWER (W, SQUIBB CORRECTED ***".

Line 163 and 164 interrogate the array L(I) where the 1000 data amplitude value is stored in the variable location V(VOLTAGE).

Each successive line performs the following algebraic operations.

Line 165 - 10 v/20-6=E

Line 166 - E2/377=P

Lines 167-168 - 1.65 PL2/4π=D

Line 169 - turns control over to subroutine "sum" which adds the contents of the variable D contents on the sum variable, I.

Line 177 returns control to line 170 which recrements the loop counter, I.

After all 1000 power amplitudes are summed, lines 171-172 causes the line printer to write out the power summation according to format statements 5 and 6 (lines 161-162).

N. LEARN

The subroutine (lines 180-184) dimensions space for 11 string variables each with 15 elements. A spectrum analyzer parameter is stored in each string variable location.

O. STATEOUT

Lines 187-207 causes the plotter to print contents of the string variable location C$ on the output plot.

DISTRIBUTION

		NO. COPIES
DRSMI-R,	Dr. McCorkle	1
DRSMI-RT,	Mr. Storey	1
DRSMI-RTS,	Mr. Daniel	1
DRSMI-RTS,	Mr. Snead	3
	Mr. Hooks	3
DRSMI-L,	Mr. Voigt	1
DRSMI-RPT,	Record Copy	1
DRSMI-RPR,	Reference	15

DATE

ILME